

IN THE CLAIMS:

1. (Original) A method for separating data blocks referenced by a writable virtual disk (vdisk) from data blocks referenced only by a backing store of a storage system, the method comprising the steps of:
 - 4 loading blocks of the writable vdisk from a disk into a memory, the loaded blocks including a writable vdisk indirect block having a plurality of fields, each field storing a valid pointer to a data block or an invalid pointer representing a hole;
 - 7 loading blocks of the backing store from a disk into the memory, the loaded blocks including a backing store indirect block having a plurality of fields, each backing store indirect block field corresponding to a field of the writable vdisk indirect block, one or more backing store indirect block fields having a pointer to a data block;
 - 11 searching each field of the writable vdisk indirect block for a hole; and
 - 12 replacing each field having a hole in the writable vdisk indirect block with a new pointer to the data block referenced by the corresponding backing store indirect block field.
1. 2. (Original) The method of claim 1 wherein the step of replacing comprises the step of:
 - 2 dirtying the data block pointed to by the backing store indirect block to enable
 - 3 write allocation of the dirty data block without altering a data content of the data block.
1. 3. (Currently Amended) The method of claim 1 wherein the step of replacing further comprises the steps of:
 - 3 | choosing a new pointer for a newly allocated data block containing ~~the-~~an unaltered data content;
 - 5 setting bits in block allocation structures for the newly allocated data block; and

6 placing the new pointer to the newly allocated data block into the field of the wri-
7 table vdisk indirect block to replace the hole.

1 4. (Original) The method of claim 3 further comprising the step of:
2 freeing the dirty data block; and
3 writing the newly allocated data block to disk.

1 5. (Original) The method of claim 4 further comprising the step of:
2 releasing an association of the writable vdisk to the backing store to thereby
3 separate the writable vdisk data blocks from the backing store data blocks.

1 6. (Original) The method of claim 1 wherein the pointers contained in the writable vdisk
2 indirect block fields and the backing store indirect block fields comprise logical volume
3 block numbers (VBNs).

1 7. (Original) The method of claim 1 wherein the invalid pointers contained in the wri-
2 table vdisk indirect block fields comprise a zero logical volume block number (VBN).

1 8. (Original) The method of claim 1 wherein the plurality of fields in the writable vdisk
2 indirect block are a writable vdisk level 1 buffer and the plurality of fields in the backing
3 store indirect block are a backing store level 1 buffer.

1 9. (Original) An apparatus for separating data blocks referenced by a writable virtual
2 disk (vdisk) from data blocks referenced only by a backing store of a storage system, the
3 apparatus, comprising:
4 a backdoor message handler adapted to load blocks of the writable vdisk and
5 backing store from disk into a memory of the storage system;

6 a writable vdisk indirect block in the memory having a plurality of fields, each
7 field storing a valid pointer to a data block or an invalid pointer representing a hole;
8 a backing store indirect block in the memory having a plurality of fields, each
9 backing store indirect block field corresponding to a field of the writable vdisk indirect
10 block, each backing store indirect block field having a pointer to a data block;
11 a special loading function for searching each field of the writable vdisk indirect
12 block for one or more fields representing a hole; and
13 a write allocator for replacing each field representing a hole in the writable vdisk
14 indirect block with a new pointer to the data referenced by the corresponding backing
15 store indirect block field.

1 10. (Original) The apparatus of claim 9 wherein the write allocator is further adapted to:
2 choose a new pointer for a newly allocated data block containing an unaltered
3 data content, set bits in block allocation structures for the newly allocated data block, and
4 place the new pointer to the newly allocated data block into the field of the writable vdisk
5 indirect block to replace the hole.

1 11. (Original) The apparatus of claim 10 wherein the write allocator is further adapted
2 to:
3 free the dirty data block and write the newly allocated data block to disk.

1 12. (Original) The apparatus of claim 9 wherein the backdoor message handler loads the
2 blocks of the writable vdisk and the blocks of the backing store during periods of reduced
3 processing activity.

1 13. (Original) The apparatus of claim 9 wherein the pointers contained in the writable
2 vdisk indirect block fields and the backing store indirect block fields comprise logical
3 volume block numbers (VBNs).

4 14. (Original) The apparatus of claim 9 wherein the invalid pointers contained in the writable
5 vdisk indirect block fields comprise a zero logical volume block number (VBN).

1 15. (Original) The apparatus of claim 9 wherein the plurality of fields in the writable
2 vdisk indirect block comprises a writable vdisk level 1 buffer and the plurality of fields in
3 the backing store indirect block comprises a backing store level 1 buffer.

1 16. (Withdrawn) A method for operating a storage system that services access requests
2 to data stored in data blocks on a storage device, the method comprising:
3 generating a read-only backing store of an organization of data blocks;
4 generating a writable image of the organization of data blocks, the writable image
5 including references to the backing store;
6 separating the backing store and the writable image;
7 deleting the backing store without interrupting the servicing of the access re-
8 quests.

1 17. (Withdrawn) The method of claim 16 wherein the step of separating further com-
2 prises the steps of:
3 searching a plurality of fields of the writable image for indications to reference
4 the backing store;
5 replacing each indication with a pointer to a newly allocated data block associated
6 with the writable image.

1 18. (Withdrawn) The method of claim 16 wherein the indications to reference the back-
2 ing store are invalid pointer values.

3 19. (Original) An apparatus for separating data blocks referenced by a writable virtual
4 disk (vdisk) from data blocks referenced only by a backing store of a storage system,
5 comprising:

6 means for loading blocks of the writable vdisk from a disk into a memory, the
7 loaded blocks including a writable vdisk indirect block having a plurality of fields, each
8 field storing a valid pointer to a data block or an invalid pointer representing a hole;

9 means for loading blocks of the backing store from a disk into the memory, the
10 loaded blocks including a backing store indirect block having a plurality of fields, each
11 backing store indirect block field corresponding to a field of the writable vdisk indirect
12 block, one or more backing store indirect block fields having a pointer to a data block;

13 means for searching each field of the writable vdisk indirect block for a hole; and
14 means for replacing each field having a hole in the writable vdisk indirect block
15 with a new pointer to the data block referenced by the corresponding backing store indi-
16 rect block field.

1 20. (Original) A computer readable medium, including program instructions executing
2 on a computer, the program instructions including instructions for performing the steps
3 of:

4 loading blocks of the writable vdisk from a disk into a memory, the loaded blocks
5 including a writable vdisk indirect block having a plurality of fields, each field storing a
6 valid pointer to a data block or an invalid pointer representing a hole;

7 loading blocks of the backing store from a disk into the memory, the loaded
8 blocks including a backing store indirect block having a plurality of fields, each backing
9 store indirect block field corresponding to a field of the writable vdisk indirect block, one
10 or more backing store indirect block fields having a pointer to a data block;

11 searching each field of the writable vdisk indirect block for a hole; and

12 replacing each field having a hole in the writable vdisk indirect block with a new
13 pointer to the data block referenced by the corresponding backing store indirect block
14 field.

Please add the following new claims:

- 1 21. (New) A method for operating a storage system comprising:
 - 2 accessing data of the storage system, the data referenced through a virtual disk;
 - 3 generating a read-only backing store of the virtual disk; and
 - 4 cloning the virtual disk by accessing data pointed to by indirect data blocks of the
 - 5 virtual disk that reference data that has been changed since generating the read-only
 - 6 backing store and by accessing data pointed to by indirect data blocks of the virtual disk
 - 7 referencing blocks of the read-only backing store that have not been changed since gen-
 - 8 erating the read-only backing store.

- 1 22. (New) The method of claim 21 further comprising releasing the backing store.